Amendments to the Claims

Please amend claims 1, 8 and 13. The currently pending claims after amendment are listed below.

1. (Currently Amended) A method for compiling computer programming code, comprising the steps of:

generating a compilable source module, said source module containing a plurality of discrete component portions;

generating selective optimization data, said selective optimization data including a plurality of selective optimization data portions, each of said plurality of selective optimization data portions corresponding to a respective component portion of said plurality of discrete component portions; and

compiling said compilable source module with an automated compiler, wherein said compiling step comprises the following steps performed by said automated compiler:

(a) with respect to each of said plurality of discrete component portions, selectively determining whether to optimize the respective discrete component portion using said selective optimization data portion corresponding to the respective discrete component portion; ~~and~~

(b) with respect to at least one discrete component portion of a first subset of said plurality of discrete component portions, said first subset containing the discrete component portions for which said selectively determining step determined to optimize the respective discrete component portion, performing at least one optimization upon the respective discrete component portion responsive to said selectively determining step; and

(c) with respect to at least one discrete component portion of a second subset of said plurality of discrete component portions, said second subset containing the discrete component portions for which said selectively determining step determined not to optimize the respective discrete component portion, compiling the respective discrete component portion without performing at least one optimization which said automated compiler has the capability to automatically perform on the respective discrete component portion.

2. (Original) The method for compiling computer programming code of claim 1, wherein said component portion is a procedure.

3. (Original) The method for compiling computer programming code of claim 1, wherein said selective optimization data comprises data concerning debug activity occurring with respect to each of said plurality of discrete component portions.

4. The method for compiling computer programming code of claim 1, wherein said selective optimization data comprises data concerning execution time with respect to each of said plurality of discrete component portions.

5. The method for compiling computer programming code of claim 1, wherein said selective optimization data comprises a plurality of optimization flags, each optimization flag corresponding to a respective component portion of said plurality of discrete component portions.

6. The method for compiling computer programming code of claim 1, wherein said compiling step comprises, with respect to a first discrete component portion, but not with respect to all said discrete component portions, generating alternative compiled versions of the first discrete component portion, wherein a first alternative version of said first discrete component portion is produced by performing a first optimization, and a second alternative version of said first discrete component portion is produced without performing said first optimization.

7. The method for compiling computer programming code of claim 1, wherein:

said step (a) comprises, with respect to each of said plurality of discrete component portions, determining a corresponding optimization level from among at least three distinct optimization levels, wherein the optimization performed at a first level are greater than the optimizations performed at a second level, and the optimizations performed at a second level are greater than the optimizations, if any, performed at a third level; and

said step (b) comprises performing optimization on each respective discrete component portion according to its corresponding optimization level.

8. (Currently Amended) A method for compiling computer programming code, comprising the steps of:

generating a compilable source module;

generating debug activity data with respect to said compilable source module; and

compiling said compilable source module with an automated compiler, wherein said compiling step comprises the following steps performed by said automated compiler:

(a) making a plurality of selective optimization determinations with respect to said compilable source module using said debug activity data; and

(b) performing at least one respective optimization step responsive to each said selective optimization determination.

9. The method for compiling computer programming code of claim 8, wherein said debug activity data comprises a plurality of counters, each counter being incremented upon the occurrence of a corresponding debug event.

10. The method for compiling computer programming code of claim 10, wherein each counter is incremented upon the occurrence of a corresponding debug event by an amount derived from a user weighting factor associated with a user on whose behalf the debug event occurs.

11. The method for compiling computer programming code of claim 10, wherein said debug activity data comprises a plurality of break-point counters, each break-point counter corresponding to a respective portion of said compilable source module, each break-point counter being incremented upon the occurrence of a break point triggered within the corresponding respective portion of said compilable source module.

12. The method for compiling computer programming code of claim 10, wherein said debug activity data comprises a plurality of variable visualization counters, each variable visualization counter corresponding to a respective variable used in said compilable source module, each variable visualization counter being incremented upon the occurrence of a user directed visualization of the corresponding variable during debug activity..

13. (Currently Amended) A computer program product for compiling computer programming code, comprising:

a plurality of executable instructions recorded on tangible signal-bearing media, wherein said instructions, when executed by at least one processor of a digital computing device, cause the device to perform the steps of:

receiving a compilable source module, said source module containing a plurality of discrete component portions;

receiving selective optimization data, said selective optimization data including a plurality of selective optimization data portions, each of said plurality of selective optimization data portions corresponding to a respective component portion of said plurality of discrete component portions; and

compiling said compilable source module, wherein said compiling step comprises:

(a) with respect to each of said plurality of discrete component portions, selectively determining whether to optimize the respective discrete component portion using said selective optimization data portion corresponding to the respective discrete component portion; ~~and~~

(b) with respect to at least one discrete component portion of a first subset of said plurality of discrete component portions, said first subset containing the discrete component portions for which said selectively determining step determined to optimize the respective discrete component portion, performing at least one optimization upon the respective discrete component portion responsive to said selectively determining step; and

(c) with respect to at least one discrete component portion of a second subset of said plurality of discrete component portions, said second subset containing the discrete component portions for which said selectively determining step determined not to optimize the respective discrete component portion, compiling the respective discrete component portion without performing at least one optimization which said computer program product has the capability to automatically perform on the respective discrete component portion.

14. The computer program product for compiling computer programming code of claim 13, wherein said component portion is a procedure.

15. The computer program product for compiling computer programming code of claim 13, wherein said selective optimization data comprises data concerning debug activity occurring with respect to each of said plurality of discrete component portions.

16. The computer program product for compiling computer programming code of claim 13, wherein said selective optimization data comprises data concerning execution time with respect to each of said plurality of discrete component portions.

17. The computer program product for compiling computer programming code of claim 13, wherein said selective optimization data comprises a plurality of optimization flags, each optimization flag corresponding to a respective component portion of said plurality of discrete component portions.

18. The computer program product for compiling computer programming code of claim 13, wherein said compiling step comprises, with respect to a first discrete component portion, but not with respect to all said discrete component portions, generating alternative compiled versions of the first discrete component portion, wherein a first alternative version of said first discrete component portion is produced by performing a first optimization, and a second alternative version of said first discrete component portion is produced without performing said first optimization.

19. The computer program product for compiling computer programming code of claim 13, wherein:

said step (a) comprises, with respect to each of said plurality of discrete component portions, determining a corresponding optimization level from among at least three distinct optimization levels, wherein the optimization performed at a first level are greater than the optimizations performed at a second level, and the optimizations performed at a second level are greater than the optimizations, if any, performed at a third level; and

said step (b) comprises performing optimization on each respective discrete component portion according to its corresponding optimization level..